AGC/VOX software for TMS320C6201† Overview Guide

www.radisys.com

World Headquarters
5445 NE Dawson Creek Drive • Hillsboro, OR
97124 USA
Phone: 503-615-1100 • Fax: 503-615-1121
Toll-Free: 800-950-0044

International Headquarters
Gebouw Flevopoort • Televisieweg 1A
NL-1322 AC • Almere, The Netherlands
Phone: 31 36 5365595 • Fax: 31 36 5365620

007-01014-0002
November 2000

November 2000
Copyright ©2000 by RadiSys Corporation.
All rights reserved.

EPC, iRMX, INtime, Inside Advantage, and RadiSys are registered trademarks of RadiSys Corporation. Spirit, DAI, DAQ, ASM, Brahma, and SAIB are trademarks of RadiSys Corporation.

† All other trademarks, registered trademarks, service marks, and trade names are the property of their respective owners.

AGC/VOX software for TMS320C6201

This document describes the AGC/VOX software functionality on the Texas Instruments TMS320C6201 Digital signal processor. The AGC/ VOX module performs level estimation, normalization of PCM samples with respect to the user specified reference level and voice activity detection, which are part of the voice processing system.

For information about installing and using the AGC/VOX software, see the topics listed below.

For information about...	Go to this page...
Installation	1
Requirements	2
Install the software	2
Compile and link the software	2
Application programmer interface	3
init_agc_vad	4
AgcMain	4
VadMain	5
Testing and performance specifications	6
Where to get more information	7

Installation

To make the AGC/VOX software fully functional, follow these steps:

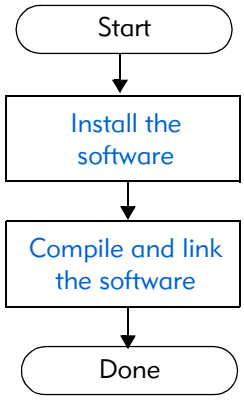

Figure 1. Installing and building the AGC/VOX software

Requirements

Before you can install this software, you need the following:

- A Microsoft† Windows† 95, Windows 98, or Windows NT† compatible PC platform.
- The following software installed on your computer:
  - TMS320C6x Code Generation Tools Release 3.01 for Windows 95, Windows 98, or Windows NT.

    —or—

    Code Composer Studio Tools, version 1.2, for Windows 95, Windows 98, or Windows NT.
  - NMAKE utility (available from Microsoft).
  - PKUNZIP utility.

Install the software

To install the AGC/VOX software:

1. Create a directory on the target system. For example:

   ```
   mkdir c:\agcvox
   ```
2. Copy the product files from the CD-ROM to the directory you just created.

Use a copy utility that retains the directory structure, such as Xcopy.

Compile and link the software

If you've already installed the TI tools, you can build the c6x version of the AGC/VOX by entering this command at the MS DOS command prompt:

```
NMAKE -f agcvc6x.mak
```

Code organization

When installed, the AGC/VOX software files reside in this directory structure:

Directory	Extension	File type	Description
\agcvox		Control source and executable for c6x	
	.agc	Output files	AGC output files.
	.c	C files	The agcvc6x.c file is a driver test program for using AGC and VOX.
	.cmd	Linker command files	The agcvc6x.cmd file controls object modules, linking options, and memory maps for the AGC/VOX module.
	.h	Header files	Header files used to build the AGC/VOX. Applications can share these files as well.
	.lib	Library files	The agcvox.lib file includes object modules used to build DSP executables.
	.mak	Build file	The agcvc6x.mak file[1] controls the build process.
	.pdf	Manual	Includes this guide, in PDF format.
	.vad	Output files	VAD output files.

[1] This file is tested to work with the Microsoft NMAKE utility, and can be modified to work with other make utilities.

Application programmer interface

Once you've installed and built the AGC/VOX software, you can use the AGC/VOX functions in your C application code.

The AGC/VOX functions accept buffers passed to the module (for example, no stack variables, and no direct access of globals).

The API for the automatic gain control module and voice-operated transmission consists of the following functions.

All module execution shall be done on 10ms frames. Channel Variable, Control, Tables buffer allocation in the data memory are not part of the mentioned APIs. This module assumes that higher-level control allocates the data memory based on the size of each structure and the alignment.

For structure variable definitions and the API usage, see agcvc6x.c.

init_agc_vad

This function is called only once. It configures appropriate parameters for the AGC/VAD

Syntax

```
init_agc_vad(
agcVar,
vadVar
)
```

Parameters

agcVar Pointer to the AGC state variable structure agcVar, size of (agcVar), 32-bit alignment, persistent memory.

vadVar Pointer to the VAD state variable structure vadVar, size of (agcVar), 32-bit alignment, persistent memory.

You must specify a target AGC reference level (agcVar->factorImg) in dBm before calling this function. Application assumes the default VAD silence threshold (vadVar->silencedB) as –60 dBm.

AgcMain

This function:

- Executes the AGC routine on *BUF_SZ* number of PCM samples.
- Runs level estimates on the new *PCMBuf* of data (agcvar->sq, agcVar->Level).
- Estimates the scale factor (agcVar->factor, agcVar->factorImg).
- Scale the *PCMBuf* and store the output in *OutBuf*.

Syntax

```
AgcMain (
PCMBuf,
BUF_SZ,
&agcVar->sq,
&agcVar->Level,
&agcVar->counter,
&agcVar->factor,
&agcVar->factorImg,
OutBuf
)
```

Parameters

agcVar Pointer to the AGC state variable structure *agcVar*, size of (*agcVar*), 32-bit alignment, persistent memory.

PCMBuf Pointer to the 16-bit PCM input buffer, 32-bit alignment, non-persistent memory (size of 2 * *BUF_SZ*).

BUF_SZ AGC block size and is fixed to 80.

OutBuf Pointer to the AGC 16-bit PCM output buffer, 32-bit alignment, non-persistent memory (size of 2 * *BUF_SZ*).

VadMain

This function:

- Runs VAD routine on *OutBuf* with *BUF_SZ* of 80. vadTbl is an initialized coefficient structure with size of (vadVar), 32-bit aligned
- Updates the thresholds and hangovers.
- Makes VAD decision using the above estimates
- Reports results such as VAD decision, *vadVar->VadFlag* in the vadVar structure

Syntax

```
VadMain(
BUF_SZ,
InBuf,
vadVar,
vadTbl
)
```

Parameters

BUF_SZ VAD block size and is fixed to 80.

InBuf Pointer to the VAD 16-bit PCM input buffer, 32-bit alignment, non-persistent memory (size of 2 * *BUF_SZ*).

vadVar Pointer to the VAD state variable structure *vadVar*, size of (*vadVar*), 32-bit alignment, persistent memory.

vadTbl Pointer to the VAD state variable structure *VadTbl*, size of (*VadTbl*), 32-bit alignment, shared memory.

Relevant structure elements

Application assumes default target AGC level (*agcVar->FactorImg*) as –24dBm, and default VAD silence (*vadVar->silencedB*) threshold as –60dBm. Application can change the above thresholds at run time between –10 to –40 dBm insteps of 6dB for AGC target level and between –25dBm to –60dBm insteps of 1dB for VAD silence threshold.

For details, see agcvc6x.c.

VadVar->VadFlag
ACTIVE_VOICE for active voice frame.

VadVar->VadFlag
INACTIVE_VOICE for silence frame.

VadVar->VadFlag
INACTIVE_VOICE_UPDATE for update frames.

Testing and performance specifications

P.56 tests on AGC c6x implementation

The c6x AGC module implementation must pass the ITU-T P.56 tests specified for measurement of active speech level. Results should match the results of the ITU-T P.56 simulation. The acceptable deviation of the active speech level should be ±3dB for the input speech levels ranging from –10 to –40 dBm. The results of the simulation should be bit-exact with the RadiSys fixed-point C simulation. These tests can be run using Texas Instruments EVM with Code Composer Studio Tools, version 1.2, to load agcvc6x.out from the agcvox directory.

Before running the executable:

- Ensure that the AGC reference (desired) output level is set in the state variable (*agcVar->factorImg* needs to be set to one of these values: –12, –18, –24, –30, –36, –42).
- Ensure that the proper path for the input/ output files is in the agcvox\agcvc6x.c file.
- Rebuild the executable using from the agcvox directory using this command:

```
nmake -f agcvc6x.mak
```

With the AGC module, you can select the desired output level between –12dBm and –42dBm in steps of 6dB.

You will find these files in the agcvox directory:

- g729a.pcm: The test vector used to test the AGC module.
- g729a.agc: The corresponding output file at –24dBm AGC level.

G.729 tests on VAD c6x implementation

The c6x VAD module implementation must pass the ITU-T G.729 Annex B tests specified to test voice activity decisions. Results should match the results of the ITU-T G.729 Annex B simulation. The percentage of acceptable deviation voice activity decisions should be < 5 % for the background noise levels ranging from –25 dBm to –60 dBm. These tests can be run using Texas Instruments EVM with Code Composer Studio Tools, version 1.2, to load the agcvc6x.out from agcvox directory.

Make sure to set the desired level, path to the I/O files specified as above in agcvox\agcvc6x.c and use this command from agcvox to generate agcvc6x.out file.

```
nmake -f agcvc6x.mak
```

Before running the executable, ensure that the AGC reference (desired) output level is set in the state variable (*agcVar->factorImg* needs to be set to one of these values: –12, –18, –24, –30, –36, –42).

You will find these files in the agcvox directory:

- g729a.pcm: The test vector used to test the VAD.
- g729a.vad: The corresponding output file at –24dBm AGC level.

Performance specifications

MCPS measurements for AGC module

The current implementation of the AGC module takes 350 instruction cycles per frame. This figure of 350 instruction cycles per (10 msec.) frame translates to about 0.035 MCPS (million cycles per second) per channel.

MCPS measurements for VAD module

The current implementation of the VAD module takes 0.88 MCPS (million cycles per second) per (10 msec.) frame.

Memory measurements for AGC & VAD modules

The following table contains memory measurements for the current c6x implementation of the AGC and VAD modules.

Module name	Program memory (Kbytes)	Data memory (Kbytes)
AGC	1.4	0.2
VAD	13.376	1.578

Where to get more information

About the AGC/VOX software for TMS320C6201

You can find out more about the AGC/VOX from these sources:

- **Readme file:** Lists features and issues that arose too late to include in other documentation.
- **World Wide Web:** RadiSys maintains an active site on the World Wide Web. The site contains current information about the company and locations of sales offices, new and existing products, contacts for sales, service, and technical support information. You can also send e-mail to RadiSys using the web site.

When sending e-mail for technical support, please include information about both the hardware and software, plus a detailed description of the problem, including how to reproduce it.

To access the RadiSys web site, enter this URL in your web browser:
`http://www.radisys.com`

Requests for sales, service, and technical support information receive prompt response.

About related products

SP6040

The RadiSys SP6040 (SPIRIT™-6040 CompactPCI† board) is a high-performance intelligent I/O subsystem designed for telecom and datacom applications. Based on

Texas Instruments† devices, the SP6040 has a 200MHz DSP engine with an I/O connector that provides interface-to-digital network interfaces (DNI) such as E1/T1 and ATM. The SP6040 contains up to four TMS320C6201 digital signal processors.

SP6020

The RadiSys SP6020 (SPIRIT-6020 board) features two Texas Instruments TMS320C6201B fixed-point DSPs that run at 200MHz and serve as the main processing engines. Each c6x can deliver up to 1600 MIPS of processing power, contingent on the available parallelism within the application code. The DSPs are used for voice and data processing, compression, and decompression.

TASK-6000

TASK-6000 software is a tool set that provides a framework for composing, executing and dynamically configuring optimized real-time TMS320C62x DSP applications. The Voice Gateway Platform (VGP), available from RadiSys, is used as the reference hardware platform.

For detailed information about TASK-6000 products, see the following publications:

TASK-6000 User's Manual, (07-0992-xx)

TASK-6000 Installation Guide, (07-1000-xx)

TI tools

Software tools from Texas Instruments used to build DSP executables.

For more information about Texas Instruments products, enter this URL in your web browser:
`http://www.ti.com/dsp`